



TAMPERE UNIVERSITY OF TECHNOLOGY

JUKKA EKLUND

**IMPLEMENTING USER INNOVATION: CASE MOBILE WEB
SERVER BETA**

Master of Science Thesis

Examiner: Prof. Ilkka Haikala (TUT)
Instructor: Jorma Hekkanen (Nokia Oyj)

Examiner and topic approved in the Information
Technology Department Council meeting on June
7th, 2006.

PREFACE

Finally, it is done. It is fair to say that this thesis has been a labor of, not love, but mostly procrastination and frustration. Since the end of 1990s I have started thesis work a couple of times over with several different titles. This work was started around 2006, and evolved over a couple of iterations into this piece mostly on Mobile Web Server in the context of user innovation. I hope you find it worth reading, and if it raises an idea or two, I am more than happy.

Thanks to the whole Mobile Web Server team and collaborators inside and outside of Nokia, and all of you who supported the cause within the company. It was a privilege to be part of something extraordinary and bit different. I got to do and learn so many different things that this thesis barely scratches the surface. Special thanks to Jorma Hekkanen for having me on the project in the first place. Last but not least, big shout to all of you who used MWS and contributed to the conversation: be seeing you!

I want to also thank the thesis examiner Ilkka Haikala for his endless patience and for the gentle but regular reminders over the years. Thanks to my mother Satu Leveelahti for coaching and cleaning the language. Not to forget my wife Suvi Syrjäläinen who gave me mental support by finishing her studies at the same time, thank you.

Tampere, May 28, 2010

Jukka Eklund

ABSTRACT

TAMPERE UNIVERSITY OF TECHNOLOGY

Master's Degree Programme in Information Technology

EKLUND, JUKKA: Implementing User Innovation: Case Mobile Web Server

Beta

Master of Science Thesis, 44 pages

May 2010

Major: Software engineering

Examiner: Prof. Ilkka Haikala (TUT)

Keywords: open innovation, user innovation, mobile application, open source, mobile device, S60, Symbian, Maemo, Mobile Web Server

Open innovation is challenging the traditional model of closed innovation in research and product development. The idea is to open the boundaries of a firm and let external influences take part in the development processes. One way of utilizing open innovation is to involve the so-called lead users via user innovation methods.

The concept of MWS "an Internet server in the phone" is an original Nokia Research Center innovation. It is a fully-fledged HTTP server that has been adapted to Nokia's Symbian smartphones. This makes it possible to use the phone features remotely from anywhere over the Internet. For the user of the phone this opens up new possibilities to use the device for example as a remote camera or data storage. If the user chooses it is also possible to open up access to the phone for friends which creates new possibilities for sharing information and communication.

The methods of open innovation were utilized in the project in many ways. First of all most of the required technology was based on open source software, such as Apache HTTP Server and Python scripting language. By employing open source software the project was able to concentrate on integration to the smartphone platform. It was also thought that using open source software components it would be easier to motivate new developers to come up with new innovative uses for technology solution.

In the MWS beta project the idea was to make the service accessible for a wider base of lead users in the mobile world. It was thought that involving lead users would lead to the discovery of new use cases for the technology. Several methods and tools for user innovation were employed in this phase of the project, such as discussion boards, a blog and direct communication channels. This thesis work documents the phases of the beta project and the results of user innovation.

TIIVISTELMÄ

TAMPEREEN TEKNILLINEN YLIOPISTO

Tietotekniikan koulutusohjelma

EKLUND, JUKKA: Käyttäjäinnovaation hyödyntäminen: esimerkkinä Mobile

Web Server beta

Diplomityö, 44 sivua

Toukokuu 2010

Pääaine: Ohjelmistotuotanto

Tarkastaja: prof. Ilkka Haikala (TTY)

Avainsanat: avoin innovaatio, käyttäjäinnovaatio, mobiilisovellus, avoin

lähdekoodi, S60, mobiililaite, Symbian, Maemo, Mobile Web Server

Avoin innovaatio haastaa perinteisen tuotekehitysmallin, joka perustuu suljettuun ajatteluun. Sen mukaan yritysten tulisi avata tutkimus- ja tuotekehitysprosessinsa ulkoisille vaikutteille. Yksi tapa hyödyntää avointa innovaatiota on ottaa niin sanotut kärkikäyttäjät mukaan kehitystyöhön.

Mobile Web Server eli ”Internet-palvelin puhelimessa” on alun perin Nokian tutkimuskeskuksessa kehitetty innovaatio, jossa täysimittainen WWW-palvelin on sovitettu toimimaan Nokian älypuhelimissa. Tämä mahdollistaa puhelimen ominaisuuksien käyttämisen miltä tahansa tietokoneelta, vaikkapa toiselta puolelta maailmaa. Puhelimen käyttäjälle tämä avaa uusia mahdollisuuksia hyödyntää puhelinta esimerkiksi kauko-ohjattavana kamerana tai tietovarastona. Käyttäjän niin halutessa hän voi myös antaa esimerkiksi ystävilleen pääsyn puhelimeensa, jolloin syntyy aivan uudenlaisia mahdollisuuksia tiedon jakamiseen ja viestintään.

Mobile Web Server -projektissa avoimen innovaation metodeja hyödynnettiin hyvin monella eri tavalla. Ensinnäkin suurin osa tarvittavasta teknologiasta perustui avoimeen lähdekoodiin, esimerkkeinä Apache HTTP Server sekä Python-ohjelmointikieli. Ajatuksena oli myös, että käyttämällä laajasti tunnettuja avoimia komponentteja voitaisiin helpommin houkutella sovelluskehittäjiä keksimään uusia käyttöjä tekniikalle.

MWS:n avoimessa beta-projektissa tuotiin konsepti laajemmin kärkikäyttäjille ja näin toivottiin syntyvän entisestään uusia ajatuksia, mihin palvelua voitaisiin käyttää. Projektin tässä vaiheessa pyrittiin käyttämään useita käyttäjäinnovaation mahdollistavia työkaluja ja toimintatapoja, kuten keskustelupalsta, blogi sekä avoin kommunikaatiokanava. Tässä työssä käsitellään beta-projektin vaiheita, miten käyttäjäinnovaatiota hyödynnettiin ja saavutetuista tuloksista.

ABBREVIATIONS

GPL	GNU General Public License
HTTP	Hypertext Transfer Protocol
LAMP	Linux Apache MySQL PHP
LAN	Local Area Network
MWS	Mobile Web Server
NAT	Network Address Translation
NRC	Nokia Research Center
OS	Open Source
OSS	Open Source Software
PAN	Personal Area Network
URL	Uniform Resource Locator
WLAN	Wireless Local Area Network
WWW	World Wide Web

Contents

Preface.....	ii
Abstract	iii
Tiivistelmä	iv
Abbreviations	v
1. Introduction	1
2. Background	4
2.1. Closed and open model of innovation	4
2.2. User innovation	5
2.3. Product management	5
3. Open Innovation.....	7
3.1. Traditional closed model of innovation.....	7
3.2. Open innovation paradigm	8
3.3. Open source software development.....	8
3.4. Benefits of the open source development model.....	10
3.5. User innovation and lead user theory	10
4. Examples of Open Innovation.....	12
4.1. Online innovation communities	12
4.2. Motivation of user innovators	13
4.3. Case example: Game industry	13
4.4. Open innovation at Nokia.....	15
5. Mobile Web Server	17
5.1. Introduction	17
5.2. Project background.....	18
5.3. Open source technology base	18
5.4. Use cases for MWS	21
5.5. Challenges for MWS	21
6. The Public Beta Project.....	23
6.1. Research background	23
6.2. Features in the beta.....	23

6.3.	First public release and feedback	25
6.4.	Marketing the beta.....	27
6.5.	Subsequent releases and other deliverables.....	29
6.6.	The end of the beta	29
7.	Implementing User Innovation	31
7.1.	Contact form: the first step.....	31
7.2.	The blog: creating a community.....	33
7.3.	Facebook application: riding with the social media trend.....	38
7.4.	MWS for Maemo: try with a new platform.....	39
8.	Summary	41
	References	43

1. INTRODUCTION

Planning of new products or improvements to existing ones is traditionally based on the knowledge about current users of the product. This knowledge can be gathered from the current customers with well-understood market research methods such as interviews. Current customers of a product however suffer from an effect called “functional fixedness”. It prevents them from seeing the product being used to anything else than what it is traditionally used. Thus studying the current users does not really give insight on how to meet the demands of tomorrow’s market. This is especially visible on fast-moving markets such as Internet and mobile services where a service or product may be born, flourish and die during the course of one year. Clearly new methods for coming up with “killer products” are needed.

Eric Von Hippel introduced in 1986 the term lead user. Lead users do not suffer from functional fixedness but instead base their views on their needs. Those needs will become general in a marketplace but only after months or years later. The definition also states that lead users will benefit strongly from a solution that addresses their needs. To put it in other words: make a solution for today’s lead users and you may have a breakthrough product when it is taken to the mainstream market. This method of user innovation has become commonplace in the Internet services market where new features or even new products can be implemented in a matter of months. Companies such as Google have pioneered this approach with services being launched in “open beta” sometimes even years before they are officially published. During the course of the beta period the users freely give out suggestions for new features or enhancements in the hope that when those are implemented they will benefit themselves. Another example of a successful lead user innovation is the open source model of software development where the lead users are individual software developers who do not just generate the new features but also submit the actual code that implements them.

Mobile phone products like any other products have been developed using traditional market research methods. During the recent years the mobile phone has evolved from a voice-oriented product into a small computer that runs a real operating

system and is extendable via user-installable applications. Hence they are called mobile devices or multimedia computers, or more generally smartphones. One of the most popular uses of a smartphone is using the Internet: browsing, email and instant messaging to name a few. Today's customer also expects that the latest smartphone can run any new web service. It is clear that the applications that run on smartphones need to be developed as fast as the Internet services evolve.

Since the mobile device has become an integral part of the Internet, Nokia has started to use open methods in developing the products for the Internet market. For example the S60 Browser is based on an open source project. New innovative applications that run on the smartphone platform are being incubated inside the company using the employees as testers and source of innovation. However there has been some hesitance to start offering these beta applications to the general public. Nokia's Beta Labs site was not opened until 2007, but has since been growing fast and currently hosts tens of new applications in varying phases of development.

Mobile Web Server is a new novel technology concept that has been born out of Nokia's research with open source software. This concept makes a mobile device accessible from the internet via a browser, just as the user would browse to a regular web site. This thesis documents how lead user innovation has been taken into use in productizing the concept and what have been the results so far. This particular project is unique since it was understood early on in the project management that somehow harnessing the lead users and open source community is going to be the only way to develop the concept further. Instead of trying to finalize the product and put it in the market, the project decided to just implement some of the essential features and put the software out in open beta. The idea was that the technology enthusiasts would take it into use and start innovating with it, maybe even coming up with a "killer application" for the technology.

The purpose of this thesis is to document the steps that have been taken so far, in order to document them for the company and benefit future projects. The information has been mainly collected through the author's experience while working on the project and interviews of key persons. Analytical data has been collected via various statistical tools available for the project and by going through the individual communication directed towards the project. In addition to documenting the outcome, some advice is given on how to further leverage open innovation methods on following projects.

The structure of this thesis is as follows. In Chapter 2 the relevant concepts of product management, open source software and user innovation are briefly explained. Also Nokia's steps so far in taking these methods in to use are presented. The open innovation paradigm and some relevant examples such as open source software development are presented in Chapters 3 and 4. The Mobile Web Server is presented in Chapter 5 to such an extent that the reader can understand why the concept is so unique. Also the history of the project is discussed to explain the choices made during the development and highlight the challenges of the chosen approach. Chapter 6 documents some of the actions taken so far in the project to tackle the challenges explained in the previous chapter. This includes the tools and methods that have been used. In Chapter 7 the outcome of actions explained in the previous chapter are analyzed. This is done by examining in depth some examples where the outcome of implementation can be shown having a clear effect, be it successful or not. Finally in Chapter 8 conclusions are drawn, giving some advice on next steps for the project and highlighting some experiences that could be utilized in the future by Nokia and mobile application development in general.

2. BACKGROUND

2.1. Closed and open model of innovation

Traditional approach to innovation inside companies is based on own research and development. This closed model is based on some hidden assumptions, such as that the important technologies can be anticipated in advance and the people working in the company's research and development are the best in the field. Evidence has shown that the closed model of innovation has many problems such as losing good ideas because of lack of resources or some internal problems with product development.

The open model of innovation introduced by Chesbrough [1] and illustrated in Figure 2.1 tackles these problems by enabling use of external innovations, such as open source software. This frees the company to concentrate on their business objectives, when everything does not need to be developed in-house (sometimes referred to as 'the not invented here' (NIH) syndrome).

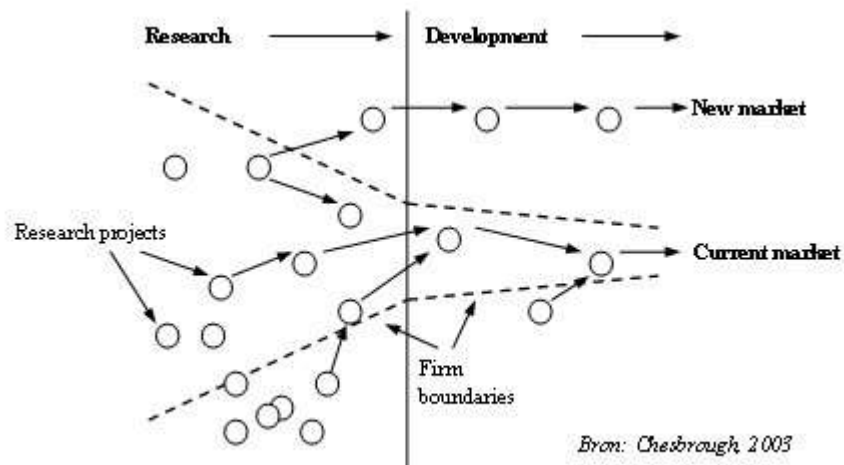


Figure 2.1: Open model of innovation [1]

One of the main motivations for this thesis is to document the execution of one particular project using open innovation methods. Open innovation is presented in more detail in Chapter 3.

2.2. User innovation

A major source for external innovation are the users of products. They know the product very well, sometimes even better than the manufacturer. Thus seeking innovation from the existing users could be a good idea. However current users of a product often suffer from a symptom called ‘functional fixedness’ [2]. They are not able to see the product used in any other way than it is used today. There is however a group of users who can see beyond today’s limitations, Von Hippel calls them ‘lead users’ [4]. They base their views on their needs instead of today’s realities. It has been shown in many real-life studies that the needs of lead users may become commonplace in the market after some time. Thus harnessing the innovation of lead users can be a very good tool for open innovation.

One particular group of lead users is open source developers who do not just have views but actually implement the product themselves by contributing code or design. For example the Linux operating system was born out of Linus Torvalds’ hobby project. One relevant open source project considering this thesis work is the Apache Foundation’s HTTP server, which runs almost half of all web sites in the world [3]. In the context of his thesis work identifying and listening to the lead users was early on identified as one of the key goals. User innovation in the context of open source development is presented in detail in Chapter 3.

2.3. Product management

Product management is a term referring to all of the different activities needed in bringing a product into market. The practical activities include planning the product and marketing the product all the way through its lifecycle. Product manager is a loosely defined organizational title for a person who is responsible for all or some of the product management activities. Depending on the company history, structure and size there can be one or multiple product managers working on one particular product or product family. Some other closely related job titles are Technical product manager, Product marketing manager and Marketing manager. [11]

There is a saying “a good product sells itself” which is of course not entirely true but it has a point: a successful product may solve a problem of a customer and hence is easy to sell. When the product manager does a good job of defining a desirable product

it makes the job of the sales department straightforward. This is where open innovation comes into play: it can help the product management finding new market opportunities. For example by listening some of the identified lead users of an existing product the product manager gets valuable information for creating a next version of the product.

3. OPEN INNOVATION

3.1. Traditional closed model of innovation

Before discussing open innovation in more detail it pays to discuss the traditional model of innovation and examine the reasons for its current problems. This traditional innovation model is called closed innovation paradigm and it has prevailed during most of the 20th century. In this model companies fund internal research facilities which invent technologies that can be then used for new products, and the resulting profit can be fed back to the research function. This is called *vertical integration* of research, and for companies with little resources for research this could mean a disadvantage in the market. Typically companies with large research functions end up with large number of innovations that never make it to the market. This is the case for example when the innovation is not considered to be in the company's core business area. Xerox is a famous example of a company that invented a number of radical innovations but failed to make successful products out of them. The Xerox Palo Alto Research Center (PARC) has been responsible for such cornerstones of personal computing as the laser printer, graphical user interface, Ethernet and object-oriented programming (Smalltalk).

During the 1990's the traditional approach of internal research started to erode due to a number of reasons, as explained by Chesbrough [15]. Academic institutions such as universities started to be involved in the commercial application of science leading to deeper co-operation with industry. The general increase in amount of skilled work-force and their mobility has led to employees changing companies more often. The availability of venture capital funding has enabled skilled researchers to capitalize on their innovations even without the help of established companies. And finally the increased amount of capable external suppliers means everything does not have to be developed and produced in-house. These factors have resulted in the market for knowledge, which does not stay within the borders of a company. Increasingly the core knowledge for any company is scattered among its employees, suppliers and universities and even its competitors and customers. Clearly the closed model of innovation based on the control of knowledge is in big trouble.

3.2. Open innovation paradigm

The open innovation model allows the knowledge to move through the company's boundaries, like a porous membrane. In practice this could mean licensing a new technology to another company if the results would otherwise end up unused. Research activities can be carried out in co-operation with universities and other organizations and the resulting innovations can be put to new products in-house or licensed outside, depending on the firm's strategy (see Figure 2.1, Page 4).

According to Chesbrough [15] a company can monetize an innovation in three different ways: making new products based on the technology, starting a new venture or spin-off company, or licensing it out to other companies. When new technology or other innovations are given out from the company there might be an entrepreneur with venture capital funding setting up a new start-up company. The numerous start-ups act as test laboratories for new products. Nowadays large companies follow closely new start-ups, co-operate with them and in some cases even acquire them. Many large companies also have their own internal venture organization, which can productize the company's own innovations faster than the core business lines.

3.3. Open source software development

One of the most well-known examples of open innovation in the technology industry is open source software (OSS) in which companies, suppliers, customers and almost everybody is collaborating on the creation of a software product. A good example of such software is the Linux operating system which started as a one-man hobby project and has since grown to be the de-facto operating system for a huge variety of computer systems.

Typical commercial software is still closed source today but actually open source ideology is closer to the origins of software production in the days of central computers in the 1970's. Back then universities and firms collaborated on software engineering in an open way: software was "free". This changed radically during 1980's when it became the norm to restrict the rights to the software source and producers started selling priced copies of software products. Regardless of the commercial success of proprietary software products such as the Windows operating system the free software ideology is alive even today in the form Free Software Foundation. Open source and free software should not be thought as the same however. Instead of the more

philosophical “hacker culture” of the free software movement, open source software highlights the practical and commercial benefits of the open software licensing models and the benefits of the collaborative development.

The most widely used open source license is the GNU General Public License (GPL), which originated in the above-mentioned Free Software Foundation. It was designed to prevent creation of proprietary closed software products out of OSS code. [16]

A typical commercial software development model follows a strict formal process including timetables, requirements and architecture of the end result. The source code of the product is kept within the company. This has the obvious implication that only quite a small amount of developers (within the company itself) are able to enhance the code and fix the errors, for example. As we have seen in earlier sections, this is in severe contrast to the open innovation thinking that has become popular during the beginning of 21st century. By giving access to the source code to a wider community, the number of potential developers of software is increased radically. Instead of a fixed number of developers within a company the software could have almost an infinite number of developers everywhere in the world.

There are however differences in the origins of open source software projects that will have an effect on the popularity of the project. In the ground-breaking paper “The Cathedral and the Bazaar” Eric Raymond [17] stipulated that open source development projects start with the needs of a person or a small group that has “an itch” that could be “scratched” with a particular piece of software. The obvious example is again the Linux operating system which was created by Linus Torvalds based on his personal interests. Linux is an example of a huge talent pool of developers working on the same project. However, an open source development model does not necessarily imply good results. In fact it has been said that most open source efforts do not succeed. They may fail in getting sufficient number of developers involved which in turn will have a negative influence on the software quality and usability, for example. In many failed cases projects are originally in-house developed and open source strategy is being applied as a means to fix problems afterwards. The appropriate use of an open source strategy from the start is the key to success. [16],[17]

3.4. Benefits of the open source development model

Having discussed the OSS model in general we can now list the potential benefits of a successful open source project. Here we assume the project goal is a commercial software offering.

All software products have errors and fixing these after the product is on the market can be very costly. With a large open source community a company gets help in the development in all stages of the software development process, before and after the market launch. The community may fix errors and even create new features. During testing phases the community can submit error reports. It will help the product get to the market faster and with better quality than as a proprietary effort. And when there are problems in the market the community will help users in public forums. All in all, open source community can help the company in almost all aspects of software development thus enabling the employees to concentrate on the critical tasks such as business and product planning.

The members of the open source community are usually also the customers of the software. Thus OSS model enables the customers to be directly involved in product development. The community works as a market research tool, for example giving direction to developing new features. Customers also benefit from this. Open source projects have to be well documented on the project web site so that customers can learn everything about the product. It is not uncommon that customers (as developers) may bring the product to a new platform such as operating system or device, when the company is not doing that. This in turn can make the product more popular in new markets resulting in more revenue for the company and more resources for the project. Along with a new platform may come also new developers and business opportunities for other products such as plug-in software and system-integration work. Involvement in open source works as a kind of marketing tool as well. Visibility on the OSS environment gives the product wider possibilities that might be impossible otherwise. [16],[17]

3.5. User innovation and lead user theory

As seen on previous sections, in the field of open source software development the role of end-users is often active as they are actually taking part in the development. Participation can be limited to testing or user-to-user support but in the other end of the

scale the user may be contributing to whole new features with source code and documentation. This is called *user innovation* and the active contributors are *user-innovators*.

The definition of *lead user* was introduced by Eric von Hippel in 1986 [4]. Lead users are defined as users of a product that are having certain requirements today, that will become commonplace in the market in the coming months or years. Lead users also benefit greatly from a solution that meets their needs. The lead user theory (also called lead user method) is a market research tool which can be used to develop breakthrough products. Instead of the traditional way of studying the current users of a product, the lead user method concentrates on identifying the lead users on the edges of the market and also outside of the current market. The lead user method has been proven in studies to be a powerful tool when used in a new product development, and the findings from lead users are in many cases also commercially attractive.

As noted in previous sections the open source developers are a special kind of lead users. In addition to identifying new innovations such as features or modifications, they will in many cases produce the innovation themselves in the form of source code. Thus the software industry is in a unique position to benefit from user innovation. This is not to say user innovation could not happen in other fields of industry. In fact the modification of products by customers is very common, but usually the results of an enhanced version remain only to benefit one user although today sharing of user innovation has become easy because of Internet and World Wide Web (WWW). Internet has also enabled the open source user-innovators to form *innovation communities*, where collective innovation is harnessed and enhanced. For example a brilliant new idea for software can be sent out by a user who is not able to contribute the code himself. Some other more capable person can then implement the feature and submit the code back to the repository. One of the largest open source communities is SourceForge which hosts over 230 000 open source projects. [16]

4. EXAMPLES OF OPEN INNOVATION

4.1. Online innovation communities

As said on the previous chapter the Internet, WWW and more recently various social media platforms such as Facebook have become one of the most visible media for discussions on new products, ideas and improvements. Researchers and companies alike are actively studying the use of online communities as sources for user innovation. The challenges with successfully benefiting from online communities are related to large amounts of different communities and participants. How the relevant communities are identified and new ideas absorbed into the innovation process of an organization?

Di Gangi and Wasko [25] have presented the potential strategies for end user innovation via online communities: user-controlled and organization-controlled methods. Organization-controlled methods include 'incorporating user toolkits' and 'integrating user innovation communities'. User toolkits are pre-determined sets of tools designed by companies to boost innovation within their existing products or services. Typical examples in the Internet are for example idea competitions where participants are asked to innovate new ways of using their products. These kind of toolkits have been proven productive and are cost-efficient to organize with the help of new social networking tools. The nature of new innovation is however limited to incremental improvements instead of radical innovation. The other organization-controlled method, integrating user innovation communities, means that company creates or internalizes an online community to be part of their innovation process. The upside with this method is potential large amount of even radical innovation, but keeping the control and being able to absorb the innovation is likely to be a challenge. One of the most well-known examples of this type of community is Dell's IdeaStorm which was started in 2007. Following the example many large technology corporations have been creating their own innovation communities.

User-controlled methods of user innovation can be divided to 'engaging lead users' and 'strategic positioning of human resources' according to Di Gangi and Wasko. Engaging lead users in online communities involves identifying, integrating and

engaging the lead users but the actual methods still remain a topic of further research today. In the Mobile Web Server beta project this challenge was tackled with various tools as explained in Chapter 7. Strategic positioning of human resources involves organization's employees taking part in external user innovation communities and then trying to integrate the potential new innovation into the R&D process. An example would be scouting relevant hobbyist discussion forums and web sites, to find out new improvements for products or ideas for new product development.

4.2. Motivation of user innovators

What motivates the lead users and innovators to work often without financial compensation? To understand this aspect of user innovation there has been several studies made. Lakhani and Wolf [27] have researched some free and OSS projects. They conclude that sense of creativity is the single biggest driver for open source programmers. In some projects the participants are actually getting paid, but that does not seem to have a negative impact on the creative side. For passionate "hackers" the OSS projects offer an outlet for their creativity. It is typical that participants in OSS projects have a full-time job producing software in a firm, but in addition they choose to use some of their free-time participating in a personally chosen OSS project. An appropriate analogy might be participation in charity work, in addition to a regular job.

Antikainen and Vääätäjä [28] studied the rewarding mechanisms in innovation communities. Most of the dedicated open innovation intermediary communities, such as InnoCentive and CrowdSpirit employ monetary rewards for top innovators. Another type of communities such as those dedicated to problem solving are not relying on monetary rewards but have various other ways of rewarding. Such other ways for rewarding are ranking lists and public acknowledgement, to name a few. We can conclude that rewarding is an important factor when considering the motivation of user innovators.

4.3. Case example: Game industry

Personal computer game industry is one of the prime examples on successful commercialization of user innovation. The game industry seems to have adopted almost all of the innovation methods described in the previous chapter such as open source software development, user innovation toolkits and utilizing online communities. Game

development as an industry is relatively young but is nowadays a major source of turnover, even more influential than the movie industry. As all games are software products, the modification and improving of end products by the end-users is in many cases possible. Since the end of 1990s modifications or “mods” of games have been elemental part of the game scene, for example in games such as Quake and Half-Life. During the 2000s the game industry has been subject of several studies on open innovation since the clear commercial success and positive embracing of the user innovation methods. Saarinen [26] has studied some of the problems associated to user innovation in the game industry.

Stickiness of information relates to the fact that exchanging information between the user-innovators and manufacturer is often slow or unreliable. This sounds strange at first but relates to the quality of information that is usable for both sender and the recipient. In the game industry’s “mod” innovation there is usually a strong dependency to the original game software and user-innovators need to use certain tools to make their contribution usable.

Intellectual property rights between original game developer and the user-innovators can be problematic, and typically game companies try to keep all of the rights to themselves. This can lead to users losing motivation to innovate if they feel themselves too exploited, for example.

Any firm needs to have a *commercial motivation* to continue supporting user innovation, and same applies to game industry since most of the game extensions created by users are free of cost. In the game industry good extensions have led to increased sales of the original game and there are even successful new products created from the user-generated mods. The *different interests and attitudes* of parties can be counter-productive to new innovation. An example from the game industry are some of the bigger game publishers whose attitude towards open innovation mods is not always favorable (possible due to many of the aforementioned reasons). On the other hand there can be many other stakeholders taking part in the innovation process in the addition of original producer and user-innovators: consultants, other publishers and various communities.

The *influence of gaming culture* on the success of user innovation has been undeniable, but there can be also negative influence with a strong culture: lack of co-operation and too influential lead users being such examples. And finally, the *composition and steady operation* of a user innovation community is a challenge.

Similar to open source communities, the success of projects often depends on just a few key contributors whose input is vital. Hence the way of creating a community and being able to control somewhat its structure are important.

These identified challenges and problems were presented in the context of game industry, but it can be argued they are valid for other fields of software industry as well where successful user innovation is sought.

4.4. Open innovation at Nokia

During the first years of the 21st century there has been increasing evidence towards open innovation paradigm being used at Nokia. The following are some relevant activities on open innovation at Nokia. All of these activities also tie in with the Mobile Web Server, as explained in more detail in the following chapters.

Formed in 1986, **Nokia Research Center** (NRC) has the mission to explore and develop technologies that will be available on the market in five to ten years. Recently NRC has been promoting the open innovation paradigm as a core method in reaching their goals. Open innovation in NRC is realized via collaboration programs with universities and partners. [10] NRC is also maintaining an “Open Source at Nokia” website that links together different OS projects and communities in which Nokia is participating. Some relevant projects concerning this thesis work are: Python for S60, Mobile Web Server, Open C and S60 WebKit. [5]

To make NRC projects and other new initiatives more visible the **Nokia Beta Labs** site was opened to the public in April 2007. First there were just a couple of applications that had originated in NRC. The idea is to release work-in-progress applications to the general public, gather feedback and thus give a public appearance for Nokia’s “beta culture” initiative. Since the launch of the site it has gained momentum gradually and has lead the way on Nokia’s initiative to becoming an “Internet company”. [8]

Forum Nokia is the developer central for Nokia products and platforms. It provides application developers with the tools and resources they need to start developing for Nokia products. There is also consulting and go-to-market services available. Forum Nokia also helps open innovation by providing the forums where developers can share their experiences and help each other. [6]

Started in 2006 the **S60 Blogs** were a collection of public web blogs centered on the S60 software platform that powers Nokia's smartphones. The author of this thesis was one of the first and most read bloggers in the site with the S60 Multimedia Blog during 2006 and 2007, and continued through 2008 with the Mobile Web Server Blog.

Maemo is Nokia's GNU/Linux-based operating system for high-end Internet devices such as Internet Tablets. The OS is based mostly on open source components. The Maemo project has a web site where developers can get all the needed resources and tools for developing Maemo applications and participate in the development of the platform itself. The platform is not licensed for other hardware vendors (like the S60 platform) and Nokia makes all of the Internet Tablet devices (such as Nokia 770, N800 and N810). Since its publication in 2005 the platform has gained lots of third-party applications. [7] More recently Maemo OS version 5 and the Nokia N900 have emerged as a more consumer-oriented offering, and the role of Maemo as Nokia's high-end device platform has been solidified.

5. MOBILE WEB SERVER

5.1. Introduction

Web server is a piece of computer software that responds to HTTP requests from a client such as a browser in a PC. Traditionally web servers have been hidden somewhere in server facilities to serve potentially millions of users at a time. Recently mobile phones have become almost as capable as PCs with a browser that is able to utilize the web pages without any special tricks such as the much-hyped WAP services in early 2000's. One could also say that current smartphones have more computing capacity than the early web servers in 1990's. With this in mind a Nokia Research Center (NRC) team decided to put a web server running on a phone. This was done partly just to show how capable a modern smartphone is, but there were also more noble goals in the background.

Even though mobile phones have become the foremost and in many areas the only method for using Internet, its role has been limited to consuming data served from remote web servers. With the Mobile Web Server the NRC team wanted to give freedom to everybody for running their own mobile web site. They even went as far as predicting that in future most personal web sites would be running on individual mobile phones, eventually replacing the traditional fixed servers as seen in Figure 5.1 (see next page). Five years after the initial project we can see that this has not happened, but who knows about the future. In the recent years we have seen the rise of cloud-computing. What if instead of fixed-server cloud farms, mobile devices could be hooked together to form a mobile cloud platform? In this scenario a mobile web server would solve many challenges. [9]

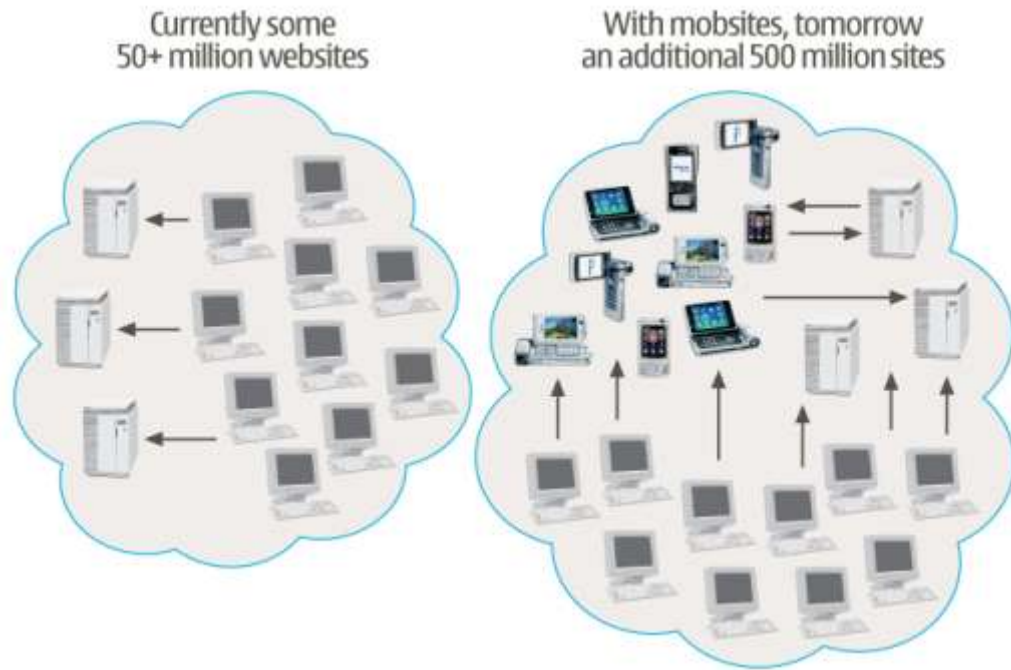


Figure 5.1: The MWS vision of mobsites [13]

5.2. Project background

As mentioned in section 2.4 the Mobile Web Server project is one of the NRC's open source projects. The project's first public release was made in May 2006, and it was actively maintained until the end of 2008. The recent focus for the open source project was to port PHP and MySQL technologies into the S60 environment, forming a complete LAMP-compliant software stack. The OS project, also called Raccoon, is intended for developers and enthusiasts but it is not easy to use for regular customers.

In this thesis the focus is on a parallel Nokia project aiming to create a more user-friendly and finalized product with the same base technology. Both projects are known with the same name Mobile Web Server (MWS), but in this thesis MWS stands for the project creating the public beta product and service. The open source version is referred to as Raccoon or open source MWS. While the projects share the same base technology the end-user version contains also closed source components and many additional features that will be covered in the Chapter 6. [9]

5.3. Open source technology base

In the NRC Open Source web site Mobile Web Server is explained as follows: "Symbian port of Apache httpd and connectivity solution that gives a mobile phone a

global URL”. It is important to note that in the MWS system there are two separate innovations linked together. Neither of them can be considered unique as similar solutions exist separately, but the combination in the context of mobile device usage is groundbreaking.

The first innovation is putting a complete web server on a mobile device. There have been earlier attempts but they have been limited to a certain subset of the HTTP stack. With MWS the ambition was to bring a fully-fledged HTTP server into the device so that there would be no difference to regular web servers running on PC or similar hardware. For this purpose Apache httpd web server was chosen. There were multiple reasons why Apache in particular was chosen. First of all it is open source with a large community around it. It is also the most used and best known web server in the world. Technically it is robust and stable, modular and designed for easy porting. In short, it is a good example of open source developed project with a thriving innovation community as explained in the previous chapter.

The chosen operating system for the mobile device in MWS is Symbian, since that was the smartphone OS for Nokia devices at that time. There were some obstacles in the porting effort but as documented in [18], the modularity of Apache and the Posix-compatible layer for Symbian made the effort a success in the end. To make the extending of the Apache possible also in the Symbian environment, the NRC team decided to port also the mod_python module since that provides access to Python scripting environment which was also ported to Symbian at the same time by another NRC team. The end result was a complete, extendable web server running on a Symbian device. The team also implemented some demo use cases which resided on the mobile device, to form a dynamic mobile site or mobsite for short.

The next problem was evident though: how can you access the web server from the Internet with a web browser? The Symbian device can connect to the Internet via cellular connectivity, but the operator’s network configuration is typically designed so that there is Network Address Translation (NAT) in effect and most likely there are several firewalls preventing access towards a mobile device. The situation is similar to the Internet connection at homes: while it is easy to set up a web server in a personal computer, without special configuration the outside world cannot access your server because of NAT configuration on the network operator side. If the NRC team had had a wireless LAN connectivity in the device they could have used that, but the use would have been very limited to the local network within reach of the WLAN access point. In

fact, they experimented with Bluetooth personal area network (PAN) but concluded that the use cases would have been too limited. They wanted to have a solution for accessing the device from anywhere in the world via Internet.

The goal for the second innovation in MWS, the connectivity solution, was not to require any modifications on the other parts of the overall system: Apache server, operator network or the browser accessing the mobile site. In addition the usage of a mobile site should be as easy as possible for a person accessing it. To achieve the goals there were two challenges to overcome: addressability and accessibility. Without a solution for addressability the mobile site would have a changing and hidden URL depending on the network configuration at any given moment. Accessibility is needed to bypass the NAT and firewall barriers. Both of these challenges were solved with a two-piece solution: a connector software running on the device and a gateway computer hosted on the Internet. All of the parts of the solution are now identified and can be seen in Figure 5.2.

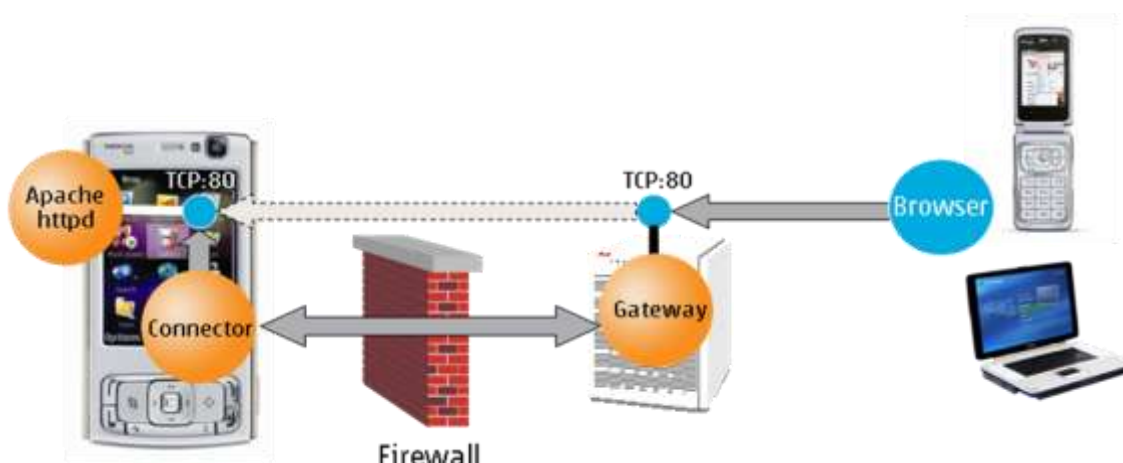


Figure 5.2: The building blocks of MWS system

The connection establishment happens from the connector to the gateway, which is crucial since the operator firewall cannot block traffic in that direction. Now there is a constant pipeline between the connector on the device and in the gateway. The communication between the two connectors is handled with a proprietary protocol. The connection is kept alive for indefinite periods of time with keep-alive messages, if there are no real data requests to the Apache server. For the web server in the device the connector is just another HTTP client, so the solution fulfills the requirement of not needing modifications in the server code. [18]

5.4. Use cases for MWS

To fully demonstrate the possibilities enabled by their solution the NRC team needed to implement some use cases as well. They were designed to show the uniqueness of the solution from two angles: How is a web site running on a mobile device different and How is a mobile device with a web server different.

Today's mobile device is not just a phone anymore but a personal storage that contains taken pictures and videos, contact entries, communication history such as SMSs. All of this information can be shared via a mobile site in real time, without the need for uploading content to a separate web service such as Facebook or Flickr. The mobile site can be interactive as well, since owner carries the mobile device usually at all times. To demonstrate this there was a "See what I see" link in the MWS default mobile site. Any visitor to the page could ask for a picture to be taken in any given moment, and it would be shown instantly on the page.

How is a mobile phone with a web server different? To demonstrate this the device owner can browse to the mobsite of his device and access the phone functions from any browser connected to the Internet. One particularly clever use for this feature would be to access the phone when it has been left behind, for example forgotten home. With MWS the user could still see received messages and interact with the device. There are also various web services where the information coming directly from a mobile device could be used automatically. NRC demonstrated this with a Google Maps application where MWS users and their location could be seen on the map.

To summarize a mobile site implemented with MWS has a number of benefits: the content is personal, interactive and context-dependent. For the device owner a web server gives a new way to interact with the device on a regular PC browser. For web services the web server is a possibility to include contextual and real-time information directly from the mobile phone and it opens up an array of possibilities.

5.5. Challenges for MWS

Having solved the major challenges with MWS there were still many aspects in the system that could prevent mobile sites from becoming widely adopted. The issues could be divided between the gateway operator and the device user. Since the solution requires a gateway computer on the Internet, somebody needs to host it and cover the costs. In the MWS project the NRC team had their own gateway set up which was open

to anybody willing to use the services. If the service would have become widely adopted the team would have faced the problem of scalability quite fast: how many simultaneous devices it can support and how is the performance scaling with those? A major issue with any web-based system is security. Similar to regular web sites a mobsite could be a target for different kind of attacks such as denial of service (DoS). To make this threat even more severe the mobsite holds very personal information of the users so any breach of security leading to stealing of private content could have serious consequences.

For the device user an active web server in the device has some practical consequences that can be severe too. The cost of data transfer falls to the device user, and this can be a bit surprising unless the user is equipped with a flat-fee service plan or uses the device via a free WLAN connection. Keeping the network connection alive and the content data served has also an effect on the battery life of the device. There was also a series of other challenges left as open questions such as how to handle the access control to a mobile site, the potential lost of privacy and performance problems to name a few. These issues would be tackled on the public beta project as explained in the Chapter 6. [19]

6. THE PUBLIC BETA PROJECT

6.1. Research background

Carlo Vainio studied in his thesis [13] the application of open source strategy as a whole in the context of Mobile Web Server. It was clearly identified that while the concept is novel and innovative, it does not immediately replace any existing solutions or generate revenue. The suggested approach was to instead promote the concept and give the necessary tools including the source code to the open innovation model and try to find the use cases together with external developers and users.

As discussed in the previous chapter the NRC Mobile Web Server was mainly research-oriented and not suitable for regular mobile phone users. Using the terminology from software engineering it could be said the solution was in a public alpha status: usable but with many issues to be solved. To address these issues Nokia made a decision to initiate a parallel project with a more consumer-oriented focus. The project goal was not to productize MWS as a product or service for a single purpose, but to continue exposing the concept to a wider public as a disrupting concept. The intention was to invite potential lead users to use the service and to develop the service together with them. This model of utilizing user innovation is common with new Internet services, where a service is often launched for public with a beta label to emphasize the unfinished status of the service. [20]

6.2. Features in the beta

With the focus clearly in a more commercially potential solution the beta project team immediately faced the challenges listed in Section 5.4. To summarize the application needed to be easy to install and use, secure, cost-sensitive and with high performance. The backend service (the gateway) needed to be scalable for potentially tens or even hundreds of thousands of users, available globally and always online. These requirements could be called enablers, and without a good solution to each the beta project could not be launched. In addition to enablers the MWS beta needed good applications for the mobile site itself. All in all, there was a lot of hard work to be done

and given the aspirations of the funding Nokia business line the project had more strict schedules and target measures for the launch. The Nokia team responsible for MWS consisted of just a couple of key persons such as a system architect, project manager, test manager and the project responsible team lead.

The initial beta features consisted of several individual applications that could be taken to use on the user's mobile site. The applications were divided into two segments: services solely for the device user and shared features for visitors to the site. The device user was given access to phone Inbox, Contacts, Calendar and Phone log. For security reasons those features could not be shared to anybody else than the device user. In addition there was an administrative section to control the web server itself and to take care of user access and other tasks. For mobile site visitors there were following applications: Blog, Camera, Gallery, Guestbook and Messaging. The reader should notice that many of the features were actually the same that were already implemented in the NRC project. However with the beta product they were made more secure and polished. The MWS team felt that with this feature set they were able to demonstrate the potential of the concept and they also had some ideas left for future implementation, possibly based on the feedback received after the launch. An example mobile site can be seen in Figure 6.1.

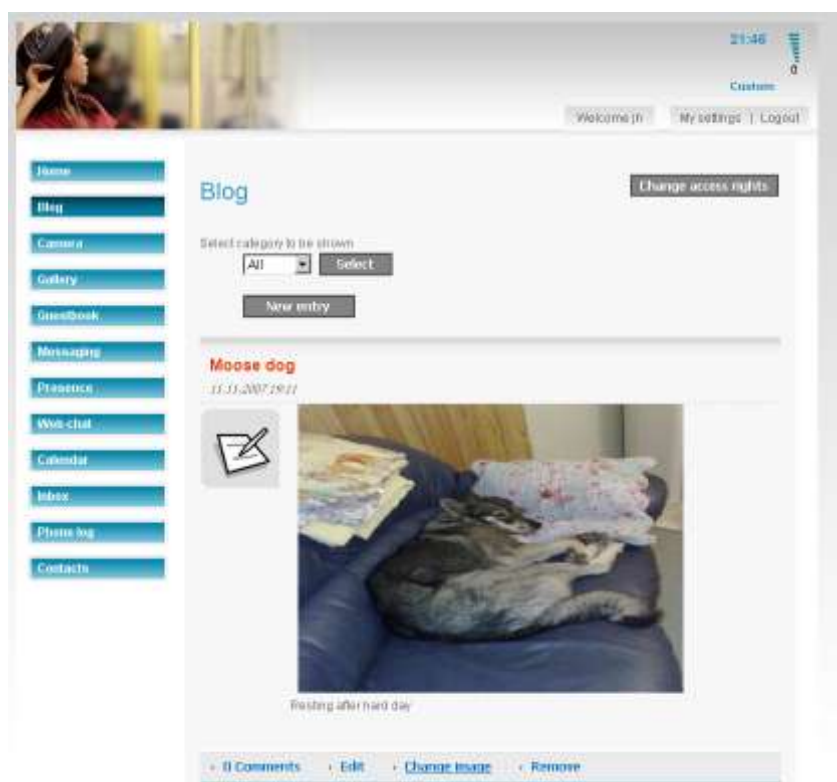


Figure 6.1: Example of a “mobsite” with a photo shared from the device

6.3. First public release and feedback

With the beta approach in mind the new Mobile Web Server offering was not brought into the market with the normal corporate way of press releases and big announcements. It was first “leaked” to a selected group of bloggers and technology enthusiasts during the MobileCampNYC [22] event in May 2007. In June 2007 the MWS 1.0 beta release was made available in its own web site (mymobilesite.net). The beta product was also highlighted at the Nokia Beta Labs giving it more visibility.

Once the initial Mobile Web Server release 1.0 was made available the team would continue the public beta track with subsequent product releases. These releases would concentrate on adding end-user features based on feedback from the users. Since the Mymobilesite.net service was not being marketed actively it was expected that the active users of the service are quite advanced technology-wise and naturally belong to the lead user segment. Initial feedback after the 1.0 release proved this assumption was to some degree correct. The feedback was collected from the public via the following tools: contact form, discussion boards at Forum Nokia and Beta Labs, Net Promoter Score and several statistical methods.

Contact form in mymobilesite.net is a simple web form to send comments or problems directly to the team. There was a dedicated person monitoring the emails and when there were non-trivial issues he would point out them to the team for further comments. This tool is analyzed in more depth in section 7.1.

Mobile Web Server Discussion board at Forum Nokia was formed already before the beta project to serve as a developer discussion board around the Raccoon project. As such it had quite active discussions on the more fundamental and technical issues. When MWS 1.0 was released the same board was used and there were certain key persons appointed to monitor the discussion and respond to the questions or comments about MWS. To date there has been over 400 discussion threads started on the forum, and it still has active conversations even though the original NRC project is not active anymore.

Beta Labs discussion forum at Nokia Beta Labs site was created “automatically” as part of the Beta Labs site renewal in the end of 2008 where all beta projects were given their own forum. Considering the active development of MWS beta ceased at about the same time, this forum will not be covered in more detail.

Net Promoter Score tool has been available since 2009 in the Beta Labs site, and also used for MWS beta since March 2009. As can be seen in Figure 6.2 this tool is remarkably easy to deploy and understand by customers and is likely to become widely adopted also at Nokia.

How likely is it that you would recommend Mobile Web Server to a friend or colleague?										
0	1	2	3	4	5	6	7	8	9	10
Not at all likely			Neutral					Extremely likely		

Figure 6.2: The NPS widget in mymobilesite.net

In addition to tools which require active contribution from the users the project has access to the usage database of mymobilesite.net web site and the connectivity gateway. The statistics available include metrics such as:

- registration: how many domains are being registered
- active domains: how many of the registered domains are kept online
- and visitors: how much traffic is being accessed on the mobile sites.

In general the feedback after the public launch was overwhelmingly positive. Below are some quotes from the feedback channel and public media:

"I love the ability to edit things on the phone with any computer. Great work though. Keep it up, and keep it free and it will go FAR."

"You guys are quick, keep up the GREAT work. It's a totally new thing. I like to call it web 3.0: 100% access to your computer, phone and internet tablets (etc) From ANYWHERE thanks to the web. And MWS is the most important/best done 'web app.' for this."

"I LOVE the Mobile Web Server, I have begun to use it for moblogging (EXTREMELY awesome, it is the first non-problematic moblogging application), gallery and more. The purpose of my MWS is to inform my friends of what I do, great when i have friends on different continents."

The single biggest conclusion after the initial “buzz” was that while people are finding the concept and its realization interesting and even ground-breaking, they are not finding a compelling enough use case for themselves. Hence the number of mobile sites being kept online was not rising with additional users. Since the whole concept of Mobile Web Server is based on the idea of the phone being reachable from the Internet at any given moment, this piece of usage data became the topmost priority and something that would keep the project busy.

The Mobile Web Server team was dedicated to keep the project evolving on the direction external feedback required. Now the practical challenge was how to prioritize the various suggestions, questions and problems presented by the users. Following tools were used during this phase:

- Roadmap, which the team had formulated already before 1.0 release. This had all of the features that had been imagined during the first implementation phase. The features that were not implemented for the first version were spread out between subsequent product releases.
- Feature request Excel sheet, which was kept updated by the person reading all feedback emails. After tackling the mundane problems, there were usually a couple of practical enhancement ideas that could be acted on.
- Individual “wish-lists”, which each member of the team kept in their own personal way.
- Error database, which was kept by the implementation team. By default all of the critical problems were supposed to be corrected so that they are available on a next release.

When the decision-making time for a new release was reached, the team would combine all available data and decide the new features or enhancements to the existing ones. This was done in a highly collaborative mode within the team with a final “seal of approval” from the project leader (instructor of this thesis).

6.4. Marketing the beta

The initial Mobile Web Server team did not have a product manager that would own the roadmap for the product and handle the marketing. The author of this thesis joined the team in September 2007 to handle the aforementioned tasks. While the initial launch was considered a success both in terms of quality and quantity of the public feedback

received, there were no on-going marketing activities happening around the Mobile Web Server. This was clearly evidenced by statistics: new registrations in to the system and usage figures were declining. Constant activity was needed around the Mobile Web Server ecosystem to keep the momentum going, so to say.

At this point the identified immediate tasks were:

- Develop the roadmap, to keep it in balance with requirements coming from the existing users and new ideas for increasing the user base.
- Handle the external marketing actions, such as forums, discussion boards, blogs and events.
- Handle the internal marketing, which is very important in a large company with possibly several competing projects being developed.

Mobile Web Server was presented in a series of internal and public events, the bigger ones being Nokia World 2007 and S60 Summit 2008. Internet advertising was tried with Google ads on two different occasions, while at the same time developing the main mymobilesite.net portal with new features and search engine optimization. There was a Mobile Web Server Blog opened under S60 Blogs community site in order to have an informal conversation forum for all parties interested in Mobile Web Server. The blog is presented in more detail in Chapter 7. The impact of these actions can be seen as the increase of visitors to the mymobilesite.net site during 2007 (see Figure 6.3). Also the decline towards the end of the year is obvious.



Figure 6.3: The increase in mymobilesite.net visitors during 2007

Why was it so important to keep the project visible constantly? Because statistics showed that the initial registered users were mainly just trying it once or twice and then forgot about it. To attract all kinds of new users and not just the “earliest of early adopters” the project needed to be visible constantly and in various media. So the top priority for external visibility was to come up with actions that would keep up the

interest also between the new releases and to seek out new ways of telling about the project to relevant parties.

6.5. Subsequent releases and other deliverables

The MWS team had laid out a target plan around the beta software releases, so that there would be a constant stream of updates to keep the interest high in the beta and to demonstrate the iterative development. The idea was to have a release during every quarter of the year. In reality the release 1.1 was made in October 2007, followed by releases 1.2 to 1.5 during 2008. In the end of 2008 it was announced that the active development of beta software has been stopped. However the connectivity gateway remained in operation, so that all users were still able to fully use their own web sites. In the next chapter there is an exploration of some of the releases in more depth in the context of open innovation.

As explained in the previous section the software releases themselves did not generate significant additional publicity for the project, even though by every release the end product was evolving to be more compelling and user-friendly. In addition to the pure marketing activities such as public events, media coverage and Google advertising the team wanted to explore other means of generating visibility as well. The Mobile Web Server Challenge was announced in March 2008 to emphasize the MWS being an ‘innovation platform’ as well. To give the developers something to work on there was also a set of Application Programming Interfaces (API) implemented in MWS release 1.3, to enable machine-to-machine communication between for example existing web services and web server in the device (mashups). As it happened the other one of the winning entries in the challenge was a Facebook application, something that the MWS team was also developing. The challenge is discussed further in Section 7.3 and The Facebook application in Section 7.4.

6.6. The end of the beta

December 2008 marked the final beta release of MWS, and also a Group Calendar widget was published. In January 2010 the discontinuation of Mobile Web Server beta was announced at Nokia Beta Labs and the hosting of mymobilesite.net gateway and service ceased. In the final announcement it was highlighted that while the end of beta service means the mobile sites cannot be reached via Internet anymore, all of the

contents that reside on the device remains on the device and can still be served for example within a local wireless network. This is a very appropriate statement as lost of personal data is often inevitable when an experimental Internet service is taken down.

7. IMPLEMENTING USER INNOVATION

7.1. Contact form: the first step

This very simple but effective tool has been in use since the first public launch of Mobile Web Server beta in June 2007. It is implemented as a web form page within the mymobilesite.net web site. The web form sends the feedback as an email to an account in the Nokia email system. This account can be accessed by all team members.

The form requires that sender gives his name, email address and the message in a free form. The page welcomes any kind of feedback as there is no formatting required. By the end of 2009 there have been a total of over 1800 emails sent via the form. The average amount of emails per month has been 60, which translates to 1 to 3 per day. This figure has remained surprisingly steady over the course of two and a half years.

The contents of the emails can be roughly categorized into the following groups:

- Technical problems (for example “I cannot install MWS, please help!”),
- questions (for example “Can I use MWS with device X?”) and
- suggestions or requests (for example “Could I share my GPS information via MWS?”).

Because of the relatively low amount of emails, the team has been able to respond to all emails personally. This has been a shared effort between the team members and the dedicated Nokia IT support help personnel. This kind of direct involvement with the end users is not common due to practical issues in large projects, but with Mobile Web Server beta it has been possible to achieve.

While all of the received feedback has been valuable, certain type of communication is more relevant considering the product development of an open innovation project such as Mobile Web Server. More specifically it is important to recognize the lead users among the customers sending feedback.

7.1.1. Recognizing the lead users

The original NRC's Mobile Web Server project was targeted towards a technology-oriented audience such as developers and academic people. As such we could state that the audience was entirely consisting of lead users. The expected contribution of lead users was to submit code to the open source project, or otherwise innovate on top of that.

With the MWS beta project the intended audience was much larger, while still assumed to be fairly technologically oriented. The new approach could be seen in the mymobilesite.net web site design which communicates the benefits of running a web server in the device in a more consumer-oriented way. By including the "Contact" web form the MWS beta team wanted to make it as easy as possible to send direct feedback. Now the challenge was how to recognize the most valuable feedback considering the iterative product development model chosen for the beta. This was done by simply browsing through the incoming feedback and identifying the original ideas among the more mundane problem statements and similar issues. Some often repeated issues did however emerge as new features too. For example the first launch "wizard" was created to make sure new users were exposed to all of the features and ended up with a fully working set of applications in their Mobile Web Server.

7.1.2. Putting the feedback to work

After the first 1.0 public release of Mobile Web Server the team had a long list of features and improvements that had to be left out from the first release because of problems such as keeping the planned schedule. Many of these features were implemented for version 1.1 in October 2007. In this release the feedback from external users cannot be seen as such, since it was mostly specified before there was a large enough user base for MWS. In hindsight it could be however seen that many of the features were "spot on" considering the received feedback. One example is the added localization for major European languages such as French, German, Italian and Spanish.

In the next planned release 1.2 the impact of end-user feedback can be more clearly seen and it was also highlighted in the communication. In addition to the already mentioned start-up wizard this major update had following new features and additions: new Messaging and Calendar applications, Access log to see visitors and short Status message field. In addition to new usage possibilities there was a "battery limit" setting

addressing the common worry visible in the feedback of running out of the device battery while keeping the MWS connected. A large number of feedback emails still consisted of error reports and those were tackled with numerous error fixes included in the release, and this practice continued throughout the project. The high quality of software in MWS project can thus be partly addressed to the open way of getting the feedback in and committing to fixing the customers' issues.

While the feedback coming from the Contact web form was valued, it also seemed to circle mostly around the more trivial issues such as technical problems. To enable a better ongoing conversation with the customers as a group there needed to be other means of communication. Given the background in public blogging of the author, creating a dedicated blog was an obvious way forward.

7.2. The blog: creating a community

Already in the beginning of Mobile Web Server project the team made sure they would directly get all the feedback generated from the public. This is not usually the case with a Nokia product, since there might be many layers of product support and similar functions between the end user and the product management. When reading themselves all feedback email and also responding to them promptly, the team was able to distill relevant feedback from the general problem statements and such. This communication was however one-to-one by nature. Many times the team would be answering to the same questions all over again because the answers could only be seen by one person at a time. After the initial public beta launch of Mobile Web Server it was seen that a public and open conversation channel would be very beneficial for the project. For this purpose a public blog was selected.

7.2.1. About corporate blogging

A blog (shortened from the original term 'web log') is a web site where a person called a blogger or a team writes articles on a regular basis, somewhat similar to diary entries. The readers of blog can usually comment on the entries so that a static article can start a conversation and serve as a basis for another article.

A corporate blog is a special kind of blog where the blogger is an employee of a company and blogging is done for business reasons. When executed well a corporate blog can have a big influence on the public perception of a company and its products.

One of the pioneers of blogging in a corporate environment Robert Scoble, then working at Microsoft, put forth a “Corporate Weblog Manifesto” in 2003 where he lays out some basic rules: [12]

- Tell the truth, the whole truth, nothing but the truth.
- Use a human voice.
- Talk to the grassroots first.
- If you don’t have the answer, say so.
- Be the authority on your product/company.

By looking at the rules one can see that this is a very different approach to traditional public relations (PR) practices of companies such as press releases, events and financial statements. These methods rely mostly on one-way communication where the company publishes something on a medium to the public. The same can be said about marketing and advertising of products and services. These traditional methods are based on strict controlling of the information coming out from a company, be it sales communication or responding to a crisis situation. Corporate blog, at least when executed based on guidelines first laid out by Scoble, changes the situation fundamentally. The blogger becomes a public face for the company or a product, and he has a direct two-way conversation with the public. For example there are no middlemen such as PR personnel between the customers and a product developer.

During this century we have seen the rise of corporate blogging pioneered by companies such as Microsoft, Sun, IBM and the like. It is notable that in the beginning the companies were mostly from the technology industry but nowadays companies in all fields of industry are running a blog or several ones. Even though open and two-way communication has its risks, clearly the benefits have been verified in many businesses.

What were then the benefits of blogging in the MWS project? The answer is two-fold: marketing and feedback. Because the MWS team was not making a commercial product there were no traditional marketing efforts being done such as advertising. The MWS product was consciously separated from any official Nokia product lines and Internet services with its own web site and identity. The evidence after the first public launch (in Internet blogs and Beta Labs) clearly showed that after the first peak of visibility new potential customers were not finding the web site. Having a blog would generate more visitors by linking the MWS to other blogs and raising the visibility of the whole concept.

The other important aspect was the open two-way and one-to-many conversation nature of a blog. While the web form email tool explained in the previous section had given a good amount of feedback, it was often quite repetitive and the team could only address one person at a time. With a blog the team could for example share the solutions to common problems to a wider audience. It was also thought that by sharing with the audience the plans for the product in advance, the potential lead users could directly contribute to product development. This kind of open sharing of planned roadmap items is usually not possible in a corporate environment. In the case of MWS the team had full ownership and independence on their product so they wanted to try that in practice.

7.2.2. Starting the blog

A practical problem when starting a blog is to find out a service provider for the blog. For MWS team the obvious choice was S60 Blogs, for a variety of reasons. S60 Blogs was already established with a wide reader base, and setting up the blog was easy and free of cost. The author of this thesis had hosted a popular blog in the site earlier which had made him well-known in the S60 community.

The MWS Blog was started in October 2007. Right from the start it was constantly one of the most read blogs within S60 Blogs, which is a good achievement given that the subject is narrower than that of most of the other blogs. The blog ran until the end of 2008, and there was a total of 48 blog posts written by the author. That is roughly a post per week which is thought to be enough to keep the blog lively. In Figure 7.1 (see next page) there is an example post on the blog.

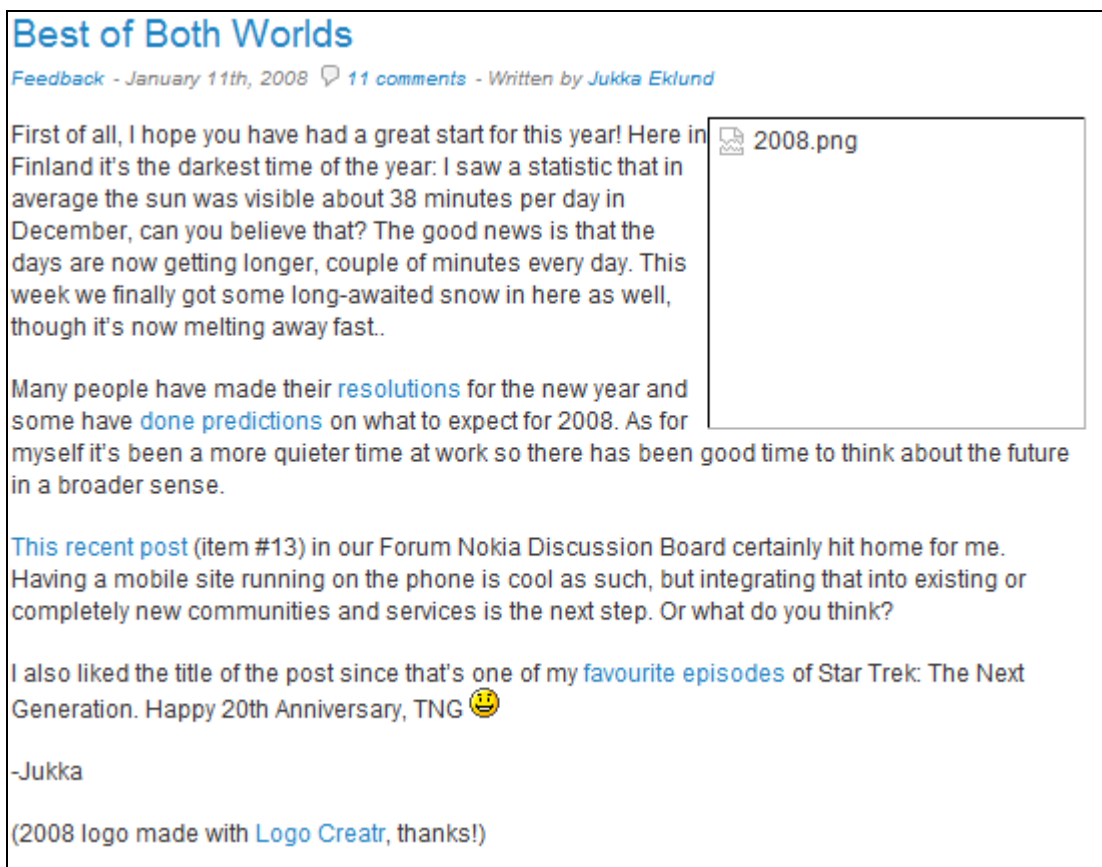


Figure 7.1: The MWS Blog main page and an example entry (see also Figure 6.2) [23]

7.2.3. The impact of the blog

By analyzing the metrics on mymobilesite.net it can be seen that the blog did not drive any significant number of new visitors to the mymobilesite.net site. For example the single biggest source of access to the site from Nokia Beta Labs generated almost ten times the traffic. One could thus say the blog did not work as a marketing tool. It is not that simple however as the impact was more indirect.

To get visitors to any web site and especially to a new one the site needs to be highly ranked in the leading search engine Google's database. Mobile Web Server suffered because it had its own Internet domain (mymobilesite.net) and a dedicated site that was initially not linked from other sites such as Nokia's. This meant low Google rank and less visitors. By having a constant stream of new content available in the blog the rank of the mother site was also raised as the domain and product was included in more Internet content. With a combination of the blog and several other actions taken to optimize the search engine visibility the Google PageRank score of mymobilesite.net site was increased to 5/10, which is a fairly good achievement for a non-commercial new site.

7.2.4. Putting the feedback to work

As explained in Section 7.1 with the beta releases 1.2 and forward there was in many cases a direct link between certain feedback and a new feature or correction in the MWS product. It was however also evident that the software releases themselves were not changing the basic problem of the beta project: there were too few active users as measured from the gateway logs. In essence there were a steady number of new users coming every month but they did not stay active, possibly because they did not see enough of value in the application for them to keep using it. We could say the Mobile Web Server was still a solution without a problem, and the attained wider visibility had not yet materialized into a single application or use case that would make it a mainstream success.



Figure 7.2: Forum post by a lead user [24]

The blog post in Figure 7.1 (page 33) mentions a forum post that is a good example of a lead user making his point about the future evolution of the mobile site towards the mainstream. The forum post is re-produced in Figure 7.2. In the forum and subsequent conversation in the blog, it can be clearly seen how some of the MWS customers were frustrated by the fact that they needed to keep the server running on the phone all the time to be reachable or to have content available for their friends and family. The

suggested approach of combining mobile server content and an existing “fixed service” was something that the MWS team tried to achieve in several ways, for example with the Facebook application explained in the Section 7.4.

7.3. MWS challenge: seeking radical innovation

Being a continuation of the NRC Raccoon project the open nature of the MWS was never completely abandoned, but it was felt that in the pursuit of mainstream use they had left the developers in the sidelines. A competition was organized to highlight the potential of MWS as an enabler for some new uses nobody had yet thought about. To motivate participants there were also modest prizes offered for the winners in the form of latest Nokia devices. The competition resulted in a variety of new ideas but there were more of academic nature than ready commercialization.

7.4. Facebook application: riding with the social media trend

One of the key assumptions of the Mobile Web Server project was that the concept is so unique that it will generate interest in the developer community. Here the relevant communities could be Apache or Python developers, but also anybody who is interested in combining the data from a MWS device into an Internet service. This kind of hybrid web development is called a mashup, Google Maps being a well-known example of such a service. Quite soon the team realized however that with the exception of a few mainly academic concepts, they needed to show-case Mobile Web Server in connection with an existing and popular web service to demonstrate the mashup capabilities of MWS. This was also highlighted in several comments from the end-users as seen in the previous section.

At that time Facebook was emerging as the most popular new social Internet service, so the team decided to implement an application for the Facebook Platform. The application could be included into the user’s Facebook page, and in combination with the MWS service running on the phone it made possible to:

- See where the user is based on GPS data, and what is he doing based on the calendar data,
- request the user to take a picture with the device camera, and show it on the page or

- send an SMS message directly to the user's SMS inbox on the device.

The MWS Facebook application was published to the site in May 2008. The peak in incoming traffic to mymobilesite.net can be seen in Figure 7.3. The brief but notable increase in visitors at mymobilesite.net can be explained by the several blog entries published by other popular bloggers following author's initial blog entry.



Figure 7.3: Increased traffic after Facebook application publication

7.5. MWS for Maemo: try with a new platform

After the initial and the few subsequent MWS beta releases there was no major evidence of developers getting excited about the prospects of building for example the kind of hybrid applications as seen in the previous section. This was somewhat a disappointment for the team, and also a slight surprise because of the unique possibilities of the system. However there is a huge amount of different application platforms “competing” to get active developers. The Facebook platform is a good example where the success of a whole ecosystem has been dependent on getting the application developers interested. When Facebook first started there were only a handful of applications available on the site, all done by Facebook itself. Today there are over 500000 active applications on the platform [14]. For a major break-through to happen an application platform needs to have a certain amount of active use or so-called critical mass. While the Mobile Web Server had a potential amount of several millions of S60 smartphone users, the installed base was only growing in a linear fashion. There were also other concerns that might have limited the developer interest. Even though the installation of the MWS system was simple and easy, there were some performance problems when developing applications due to the relatively heavy combination of web server coupled with the Python framework. This was also noticed while developing the Facebook application.

Taking above into consideration the team wanted to explore a parallel track with Nokia's Maemo device platform. This was seen as something fairly easy to achieve

since Apache web server and Python had been already ported for the Linux-based platform by the open source community. The only thing missing was the connectivity client for Maemo. This was developed as part of MWS beta release 1.3 but remained in an internal alpha stage and was not made publicly available.

The Maemo release was again taken up during 2009 as Nokia released a significantly enhanced version of the platform in form of Maemo 5 and the Nokia N900 device. The MWS enabler for Maemo was further developed to work with this device and there were also a couple of demo cases built to highlight the web server functionality.

8. SUMMARY

The purpose of this thesis was to document the steps taken in the Mobile Web Server beta project, from the initial public offering until conclusion which in this case was closing of the service without, at least immediate, commercialization. The theoretical background for the work is open innovation, user innovation and open source software development in general.

Mobile Web Server is a novel technology concept created in the Nokia Research Center. According to the research direction the technology was made publicly available from the start for the mobile enthusiast and open source community. In addition to the research team, Nokia wanted to test the concept with regular end users. For this to happen the research-oriented and somewhat cumbersome system was developed into an easier to use experimental service. However the product lacked a clear focus and use cases, and the idea was to refine the concept together with the users. To gain the needed visibility for the service Nokia Beta Labs was initially used as the marketing channel. The MWS team continued to develop their own web site as well, adding new features and content.

Since involving the users in the development was crucial there needed to be communication tools to interact with the users and also to make offering more widely-known. A number of tools were chosen, based on resource constraints and expected match to find the relevant users for the product. The existing “Nokia enthusiast” community was utilized with a dedicated blog to give the project its own voice and to influence the users. A simple contact form in Mobile Web Server portal generated the most feedback, but quality of feedback was not consistent with often repetitive complaints about technical problems.

Mobile Web Server beta suffered from the lack of “killer application” throughout its existence. The initial offering was centered on the concept of user’s own mobile web site. While that was a logical starting point inherited from the research project it also narrowed the scope of new ideas being generated. Looking at the evidence it can be said the beta project was successful in implementing new enhancements to the system based

on collaborative user innovation mode. However the concept itself would have needed a radically new and innovative use for the existing technical solution, something that would have made the technical challenges less visible. The team could not overcome a fundamental problem: direct communication between devices required the MWS to be installed on each device and more importantly it needed to be constantly connected to the Internet. Based on statistical information this was not achieved with the existing user base. A constant Internet connection has some severe drawbacks for the user, increased battery consumption and costs of network data being the major ones.

Further research or product development utilizing MWS concept should first tackle the barriers for use in the environment: the device platform, network connectivity and business environment. A potential choice for further development is the Maemo (recently renamed to MeeGo) platform. The MWS solution network traffic and battery consumption should be minimized. To enable seamless solution the Mobile Web Server should be an integrated part of the device OS, and the application development should be based on open APIs. As a business environment the MeeGo ecosystem should be studied further. As an integrated part of the open source OS the Mobile Web Server concept would also benefit from the established developer community and tools.

In this thesis work user innovation methods were proven to be effective to a certain degree. The project team was able to identify some of the needed enhancements and usability challenges in a fast iterative fashion. There was also a feeling of community created. The common problems with user innovation were however present throughout the project: how to find the right participants, motivate them and enable radical innovation via online tools.

The selected online tools did not fully utilize the possibilities of available technology, especially considering the rewarding and motivation of participants. This can be partly explained with the limited resources, the team were in many cases forced to use the tools available already by the company. In further work utilizing some of the existing innovation communities should be considered, to address both the quantity and motivation of the user innovators.

REFERENCES

- [1] Chesbrough, Henry. *Open Innovation: The New Imperative for Creating and Profiting from Technology*. Boston, USA, Harvard Business School Publishing Corporation, 2003.
- [2] Adamson, R. E. Functional fixedness as related to problem solving; a repetition of three experiments. *Journal of Experimental Psychology* 44, pp. 288–291, 1952.
- [3] Web Server survey [web site].
http://news.netcraft.com/archives/web_server_survey.html. Visited 2.1.2010.
- [4] Von Hippel, Eric. Lead users: a source of novel product concepts. *Management Science* 32, pp. 791–805, 1986.
- [5] Open Source at Nokia [web site]. <http://opensource.nokia.com>. Visited 20.2.2009.
- [6] Forum Nokia [web site]. <http://forum.nokia.com>. Visited 20.2.2009.
- [7] maemo.org [web site]. <http://maemo.org>. Visited 20.2.2009.
- [8] Nokia Beta Labs [web site]. <http://betalabs.nokia.com>. Visited 20.2.2009.
- [9] Mobile Web Server [web site].
<http://research.nokia.com/research/projects/mobile-web-server/>. Visited 20.2.2009.
- [10] Nokia Research Center [web site]. <http://research.nokia.com>. Visited 2.1.2010.
- [11] Johnson, Steve. *The Strategic Role of Product Management* [e-book].
- [12] The Corporate Weblog Manifesto [web site].
<http://radio.weblogs.com/0001011/2003/02/26.html>. Visited 2.1.2010.
- [13] Vainio, Carlo. *Application of Open Source Strategy on Mobile Web Server*, Master of Science thesis. Helsinki, Helsinki University of Technology, 2007.
- [14] <http://www.facebook.com/press/info.php?statistics> [web site]. Visited 2.1.2010.
- [15] Chesbrough, Henry, Vanhaverbeke, Wim & West, Joel. *Open Innovation: Researching a New Paradigm*. Oxford, Oxford University Press, 2006.
- [16] Von Hippel, Eric. *Democratizing Innovation*. USA, MIT Press, 2005.
- [17] Raymond, Eric. *The Cathedral and The Bazaar*. USA, O'Reilly Media, 2001.
- [18] Wikman, Johan & Dosa, Ferenc. *Providing HTTP Access to Web Servers Running on Mobile Phones*. Nokia, 2006.
<http://research.nokia.com/files/NRC-TR-2006-005.pdf>
- [19] Wikman, Johan, Dosa, Ferenc & Tarkiainen, Mikko. *Personal Website on a Mobile Phone*. Nokia, 2006. <http://research.nokia.com/files/NRC-TR-2006-004.pdf>
- [20] Hekkanen, Jorma. Discussions about MWS project phases. December 2009.
- [21] Pusa, Juha. Discussions about MWS project. December 2009.
- [22] <http://barcamp.org/MobileCampNYC> [web site]. Visited 20.2.2009.

- [23] <http://discussion.forum.nokia.com/forum/showpost.php?p=375907&postcount=13>. Visited 20.2.2009.
- [24] Mobile Web Server Blog. [web site]. Not accessible anymore, last visited 20.1.2010.
- [25] King, William R. (editor). Knowledge Management and Organizational Learning, pp. 199–205. USA, Springer, 2010.
- [26] Saarinen, Hannu. Käyttäjänovaatioiden mahdollisuudet ja ongelmat – tapaustutkimus peliteollisuuden käytännöistä. Finland, Helsinki School of Economics, 2009.
- [27] Lakhani, Karim R., Wolf, Robert G. Why Hackers Do What They Do: Understanding motivation and Effort in Free/Open Source Software Projects. Perspectives on Free and Open Source Software. USA, MIT Press, 2005.
- [28] Antikainen Maria J., Väättäjä Heli K. Rewarding in open innovation communities – how to motivate members. International Journal of Entrepreneurship and Innovation Management 2010, Vol. 11, No. 4, pp. 440–456.